ConsensusClustering Documentation

Module name: ConsensusClustering
Description: Resampling-based clustering
Author: Stefano Monti, Marc-Danie Nazaire (Broad Institute), gp-help@broad.mit.edu
Date: 3/23/2007
Release: 2.0

Summary: Given a set of items to be clustered (items can be either genes or chips/experiments), ConsensusClustering (CC) provides for a method to represent the consensus across multiple runs of a clustering algorithm and to assess the stability of the discovered clusters. To this end, perturbations of the original data are simulated by resampling techniques. The clustering algorithm of choice is applied to each of the perturbed data sets, and the agreement, or *consensus*, among the multiple runs is assessed and summarized in a *consensus matrix.* Each matrix entry is indexed by an item pair, and it measures the proportion of times the pair's items are clustered together across the resampling iterations (ideally, always, or never). A distinct consensus matrix is generated for each of the number of clusters considered (e.g., if kmax=5, consensus matrices corresponding to 2, 3, 4, and 5 clusters will be generated). Visual inspection of the consensus matrices, and of the corresponding summary statistics can be used to determine the best number of clusters (see reference for more details).

References:

- S. Monti, et al. "Consensus Clustering: A resampling-based method for class discovery and visualization of gene expression microarray data", *Machine Learning Journal*, 52(1-2):91-118, 2003.

Parameters:

Name	Description	Choices
input.filename	The data based on which to carry out the clustering	It can be a '.gct', a '.res', or a '.odf' file.
niter	Number of resampling iterations	Positive integer
kmax	Try K=2, 3, ..., kmax clusters	1 < kmax < number of items
normalize.type	Type of normalization to perform on data	1=row normalize; 2=column normalize; 3=both (default);
norm.iter	Number of row/column normalization iterations to perform. It supercedes normalize.type.	A non-negative integer
seed.value	Seed for random number generator	Default: 12345
algo	Clustering algorithm to use	HIERARCHICAL; SOM; NMF; KMEANS
cluster.by	Whether to cluster by rows/genes	rows;columns

	or columns/experiments	
distance.measure	Distance Measure	EUCLIDEAN(valid only for Hierarchical and Kmeans); PEARSON (valid only for Hierarchical)
resample	resampling scheme to use	'subsample[ratio]', 'features[nfeat]', 'nosample'
merge.type	How to update the distance measure (ignored if algo is other than 'HIERARCHICAL')	"average", "single", "complete".
descent.iter	Number of SOM/NMF iterations (ignored when algo is hierarchical):	positive integer (default: 2000)
pink.size	point size of a consensus matrix's heat map	$1 \leq$ pink.size ≤ 20
out.stub	Stub pre-pended to all the output files	
create.heat.map	Whether to create heat map images (one for each cluster number)	yes/no (default: no)
heat.map.size	point size of a consensus matrix's heat map	$1 \leq$ heat.map.size ≤ 20

Return Value:

1. <out.stub>.<sampleid>.<k>.clu, is a text file listing the items belonging to each cluster. (<sampleid> indicates the type of resampling scheme, and <k> denotes the number of clusters).
2. <out.stub>.<sampleid>.<k>.gct is the consensus matrix for <k> clusters, with the entries sorted as in the input data.
3. <out.stub>.<sampleid>.srt.<k>.gct is the consensus matrix for <k> clusters, with the entries sorted so as to have items clustering together adjacent to each other.
4. <out.stub>.<sampleid>.srt.<k>.gif is the heat map corresponding to the sorted consensus matrix.
5. <out.stub>.<sampleid>.statistics.pdf includes a series of plots of statistics (Lorenz curve, Gini index, Consensus CDF) that can be used to determine the best number of clusters.

Platform dependencies:

Task type: clustering
CPU type: any
OS: any
Java JVM level: 1.4
Language: Java, R (including the R library 'ineq')
Support files: none